Mining Building Metadata by Data Stream Comparison

Holmegaard, Emil; Kjærgaard, Mikkel Baun

Published in:
Proceeding of the 2016 IEEE Conference on Technologies for Sustainability

Publication date:
2017

Citation for pulished version (APA):
Holmegaard, E., & Kjærgaard, M. B. (2017). Mining Building Metadata by Data Stream Comparison. Proceeding of the 2016 IEEE Conference on Technologies for Sustainability, 28-33.

Mining Building Metadata by Data Stream Comparison

Emil Holmegaard and Mikkel Baun Kjærgaard
SDU - Center for Energy Informatics
Mærsk McKinney Møller Institute
University of Southern Denmark
Email: {em,mbkj}@mmmi.sdu.dk

***Abstract*—Improving at scale the energy performance of buildings requires that applications are portable among buildings (i.e. the same application in two different buildings). One challenge in enabling portable applications is metadata about building instrumentation. The problem is that there are multiple ways to annotate sensor and actuation points. This makes it difficult to create intuitive queries for retrieving data streams from points. Another problem is the amount of insufficient or missing metadata. We introduce Metafier, a tool for extracting metadata from comparing data streams. Metafier enables a semi-automatic labeling of metadata to building instrumentation. Metafier annotates points with metadata by comparing the data from a set of validated points with unvalidated points. Metafier has three different algorithms to compare points with based on their data. The three algorithms are Dynamic Time Warping (DTW), Empirical Mode Decomposition (EMD), and the differential coefficient. Two of the algorithms compare the slope of the data stream in the values. EMD finds similarities based on the frequency bands among the data stream. By using several algorithms the system is robust enough to handle data streams with only slightly similar patterns. We have evaluated Metafier with points and data from one building located in Denmark. We have evaluated Metafier with 903 points, and the overall accuracy, with only 3 known examples, was 94.71%. Furthermore we found that using DTW for mining points with the point type of room temperature achieved an accuracy as high as 98.13%.**

I. Introduction

The amount of interconnected devices today is more than 22 billion which is expected to reach 50 billion devices by 2020 [8]. A lot of Internet of Things (IoT) devices are appliances for giving more comfort to the residential environment and all kinds of sensors and actuators used for building instrumentation to increase the level of automatization in buildings. Building instrumentation covers the sensor and actuation infrastructure within the building. There are today no common language for communication between IoT devices. For many IoT applications there is a need for data-on-demand from IoT devices which will be retrieved using sophisticated intuitive queries [7]. Those queries have to be based on metadata e.g. from the building instrumentation, to query based on the context of the point. A *point* represents a connection between the cyber and physical world which may be discretized into a data stream. Such data streams contains either sensor readings or actuation requests depending on direction. For example could a query be "find all points handling temperature in room y", which returns actuators and sensors for room temperature in room y. However, without a common metadata language different devices will not be able to interpret data from other devices.

To reach the energy goals for buildings in the near future, multiple solutions have to play together [10]. This will require that applications are portable among buildings. By portable we mean, having applications that can be used in multiple buildings with a minimal effort of installation. For having a portable building applications, there will be a need for simple but efficient way to fetch data from points. A level of abstraction for fetching data, could be fetching a certain point, by only knowing the point type and location. Exposing a building instrumentation with an API for enabling control on top of it, is what we will call a Software Defined Building. There will be a need for annotated metadata for all points in buildings to enable Software Defined Buildings. The challenge with metadata is that there are multiple ways to annotate sensor and actuation points [2]. This makes it difficult to create intuitive queries, that can port to multiple buildings, for retrieving data streams from points and thereby make a proper level of abstraction [9]. Another problem is to maintain or set up the building instrumentation which require annotating metadata for the points. Calbimonte et al. [4] argue that metadata will be held at a very low level or be incorrect if the person whom annotate the sensor does not have to benefit from the metadata.

To address these challenges we introduce Metafier, a tool for extracting metadata from data streams. In Metafier the points have a state of either validated or unvalidated, where validated points has a correct set of metadata and a validation of corresponding data stream. This gives us the option to compare unvalidated points with the validated points, and then transfer the relevant metadata to the unvalidated point, if the similarity meets a certain threshold.

The contributions of the paper is:

- Based on minimal set of 3 validated points, we demonstrate that our algorithms are enable to estimate point type with an accuracy of 94.71%
- Using DTW for mining metadata, we are enable to estimate the point type of room temperature with an accuracy as high as 98.13%.
- Metafier can semi-automate the task of annotating building instrumentation

II. Approach

In this section we will explain the used approach in Metafier. The system consists of a GUI and a API. Metafier does not store any sensor readings, but has a interface for the simple Measurement and Actuation Profile (sMAP) proposed by Dawson-Haggerty et al. [5]. sMAP has a REST API for interacting with two databases, one for time series data (data streams), and one for metadata regarding the time series data. Metafier retrieves data streams from the time series database and store annotations for points in the metadata database.

Fig. 1 illustrates the flow within Metafier. We assume that we have a set of validated and unvalidated points. Circles indicate validated points, rhombuses indicate unvalidated points. The three colors of the circles and rhombuses indicate the different point types, the green color could indicate a point with the type of room temperature sensor. A validated point, is a point which has been marked with a flag, indicating that metadata for this point is correct. Furthermore the data stream for this point has been validated and match the correct point type, e.g. a point with the type of room temperature sensor that has a data range between 15°C and 30°C. The blue arrows indicate tasks performed by Metafier. The system retrieves data streams from a sMAP installation. Metafier run the algorithms for each data stream and calculates the similarity confidence of whether the data streams are similar. The similarity confidence is calculated based on e.g. the cost matrix from Dynamic Time Warping. The green arrows indicate tasks performed in Metafier GUI by a user whom has knowledge of the building instrumentation. The role of a user is to evaluate whether the similarity confidence has reached an acceptable threshold and select whether metadata should be transferred or not.
The three algorithms in Metafier are DTW, EMD, and the differential coefficient. The three algorithms compare the slope of the values in the data streams. All of the algorithms creates an estimate, which has a similarity confidence of how similar the unvalidated point are to the validated point.

A. Slope Compare

Slope Compare divides the data stream into smaller chunks of 4 sensor readings. The algorithm then finds whether the data stream decrease or increase more than $\pm$ 0.05. This value can be changed based on the expected point type, but has not been changed for this setup. The algorithm calculates the differential coefficient by taking the simple formula for finding slope between two points. The change of the slope is calculated as $\frac{y_n - y_1}{x_n - x_1}$, where y is the value of the sensor reading, x is the index, 1 indicates the first coordinate, and n the last coordinate of the chunk. The method returns the midpoint of the calculated slope. After computing all the indices with a slope change, we compare the result for the two streams. We have used a list comparison as described in Section II-D.

B. Empirical Mode Decomposition

This algorithm follows the same approach as Fontugne et al. [6] for Empirical Model Decomposition. Instead of finding anomalies, we use the signature for finding similarities in the data streams. The algorithm decomposes a data stream of a point into an additive set of components called Intrinsic Mode Functions (IMF). IMF identify patterns of the data stream in different frequency bands. The IMF are afterwards aggregated, where we remove high frequencies, which are all IMF with a time scale lower than 20 minutes. When we have the aggregated IMF of a point, we compare the point with a EMD result of a validated point. For calculating the similarity confidence of similarities between two points, we use a list compare as described in Section II-D.

C. Dynamic Time Warping

Dynamic Time Warping (DTW) is used for measuring similarities between two temporal sequences which may vary in speed. The data streams from points are indeed temporal sequences, e.g. the effect of sun light in two rooms can vary in speed. DTW calculates a cost matrix, giving the distance between the two data streams. DTW also finds the shortest path, based on the cost matrix. Besides the cost matrix we have compared mean, max and min of the two data streams. If the two data streams have a low cost, but have huge difference in magnitude, our implementation will set the data streams to be different. We have used a range of $\pm 10\%$. For DTW the similarity confidence has been calculated based on the cost matrix.

D. List Similarity

For comparing of lists, we have used a combination of length similarities and cosine similarities. The length similarity is given by $ls = \frac{min(len(A),len(B))}{max(len(A),len(B))}$ and is the difference between lengths of list A and list B. The length similarity gives a value between 0 and 1. The cosine similarity, cs, is given by $cs = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$ where A is the result of the validated point and B is the result of the unvalidated point, both given as vectors. The cosine similarity gives a value between 0 and 1 for how similar the orientation of the vectors are, not how similar the magnitude is. Two vectors with a cosine similarity of 1 has the same orientation. If the vectors are perpendicular oriented, the cosine similarity will be 0. The cosine similarity gives a value between 0 and 1 for a positive space.

III. Evaluation Setup

We have evaluated the algorithms in Metafier, using data from the Green Tech Center[1] building. The building is from 2014, it is instrumented with 903 points. The building is 3000 m^2 and consists of 50 rooms spread over three floors. The building is located in Vejle, Denmark and is an office building. We have chosen a set of 9 rooms, 3 at each floor, we have selected 4 offices and 5 conference rooms. For all rooms we have validated points representing room temperature, CO_2

[1] http://greentechcenter.dk

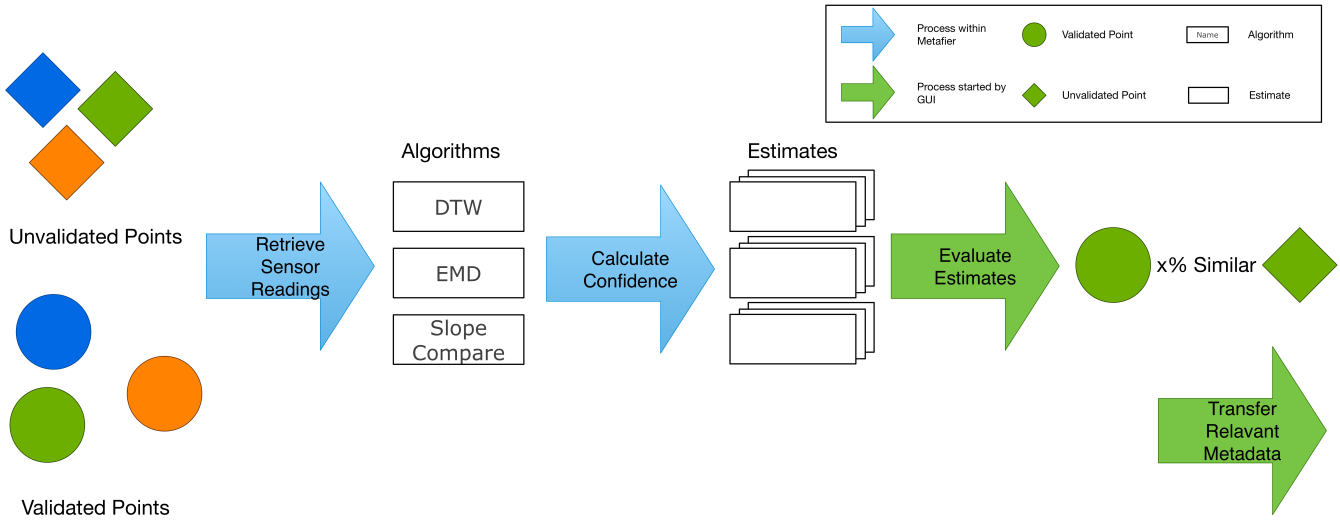

Fig. 1: The flow in Metafier. The blue arrows indicate processes within Metafier. The green arrows indicate processes started from the GUI.

level, and llluminance. This gives 9 rooms with 3 points in each, which is a total set of 27 validated points. All 903 points have been manually labeled with point type, as ground truth. The data from all points in the building are sampled with an interval of 5 minutes, and we have used data from 7 days (Midnight June 28, 2016 to Midnight July 5, 2016). The effect of occupancy have not been taken into consideration. We have used a similarity confidence over 75%, as this is the acceptable threshold which the user would have used. All estimates with a similarity confidence over 75% are used as a true positive, when both validated and unvalidated have the same point type. Accuracy is calculated as $\frac{TP+TN}{TP+TN+FP+FN}$, where TP are true positives (i.e. a unvalidated point with the same point type as the validated point and a similarity confidence over 75%), TN are true negatives, FP are false positives, and FN are false negatives. For evaluation of the selection of validated points, all experiments have been created as one-by-one for each of the validated point and tested for all other points. Afterwards we have created different combinations of the validated points. We have created all combinations with 1, 3, 5, and 7 of each validated type. For the combination of 1, we have used one validated point with the point type of room temperature, one CO_2 level, and one llluminance. For the combination of 3, we have used three validated points with the point type of room temperature, three CO_2 level, and three llluminance. For the combination of 5, we have used five validated points with the point type of room temperature, five CO_2 level, and five llluminance. For the combination of 7, we have used seven validated points with the point type of room temperature, seven CO_2 level, and seven llluminance. For the combination of 7, we have used 21 of the 29 validated points.

IV. Results

This section present the results from the for algorithms in Metafier, Empirical Mode Decomposition, Dynamic Time Warping, and Slope Compare. Furthermore a combination of the three algorithms are shown under the label All, for all figures Fig. 2 to 5 . As described in Section III, combinations of validated points have been shown in Fig. 2 to 5. The results have been split for each point type. The x-axis shows the point type and the algorithm which have produced the result. The y-axis shows the accuracy in percent. For llluminance the box is colored blue, for CO_2 level the box is colored green, and for room temperature the box is colored red.

The results in Fig. 2 for llluminance shows a minimum accuracy of 94.49% and a maximum accuracy of 94.71% with a Standard Deviation (SD) of 0.06. For CO_2 level the minimum accuracy was 94.60% and a maximum accuracy of 97.19% with a SD of 0.53. For Temperature the minimum accuracy was 94.38% and a maximum accuracy of 98.13% with a SD of 1.22. For DTW the maximum accuracy was 98.13% for Temperature. For Slope Compare the maximum accuracy was 97.14% for Temperature. For EMD the maximum accuracy was 97.19% for CO_2 level.

The results in Fig. 3 for llluminance shows a minimum accuracy of 94.49% and a maximum accuracy of 94.64% with a SD of 0.04. For CO_2 level the minimum accuracy was 94.60% and a maximum accuracy of 95.78% with a SD of 0.25. For Temperature the minimum accuracy was 94.49% and a maximum accuracy of 98.09% with a SD of 1.10.

The results in Fig. 4 for llluminance shows a minimum accuracy of 94.49% and a maximum accuracy of 94.60% with a SD of 0.03. For CO_2 level the minimum accuracy was 94.60% and a maximum accuracy of 95.46% with a SD of 0.22. For Temperature the minimum accuracy was 94.54% and a maximum accuracy of 98.01% with a SD of 1.08.

The results in Fig. 5 for llluminance shows a minimum accuracy of 94.49% and a maximum accuracy of 94.57% with a SD of 0.03. For CO_2 level the minimum accuracy was 94.60% and a maximum accuracy of 95.26% with a SD of

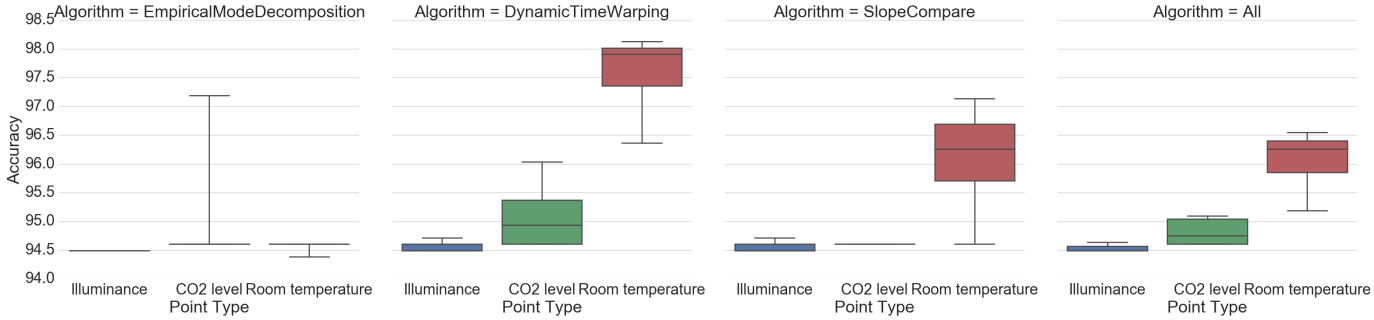

Fig. 2: Results with a combination of 1 (3 validated points) for the three algorithms and All.

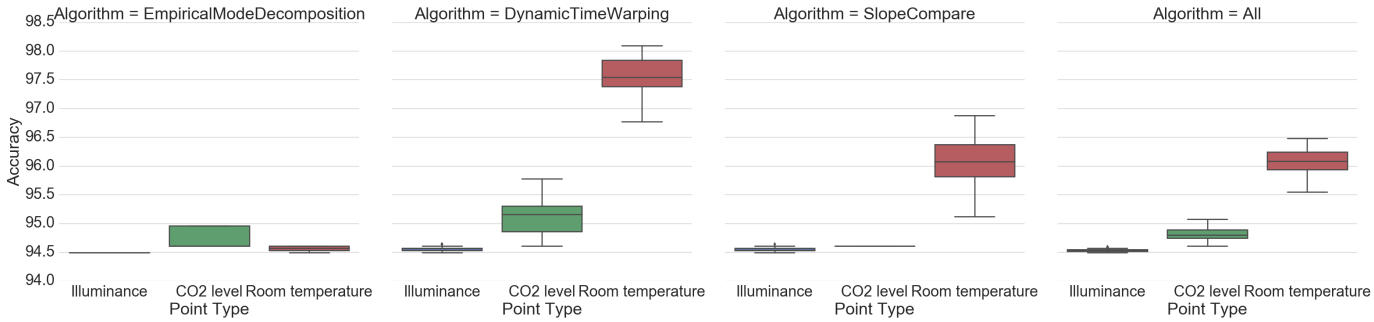

Fig. 3: Results with a combination of 3 (9 validated points) for the three algorithms and All.

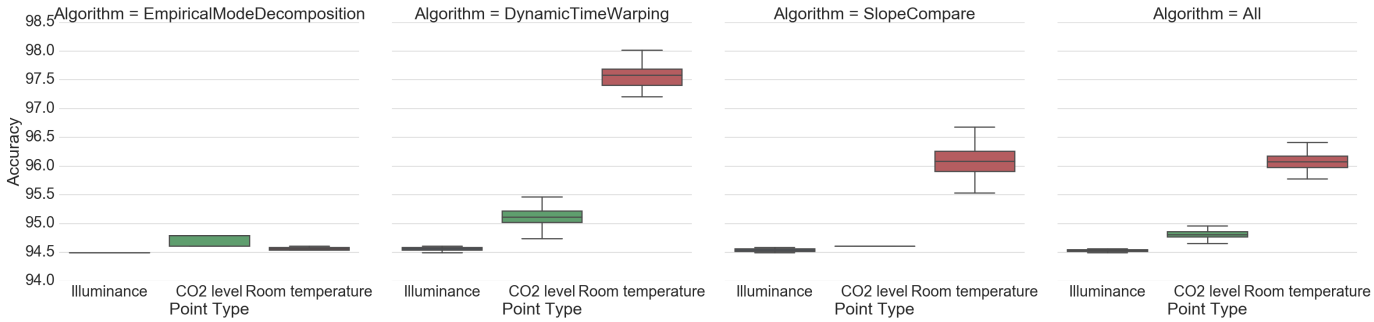

Fig. 4: Results with a combination of 5 (15 validated points) for the three algorithms and All.

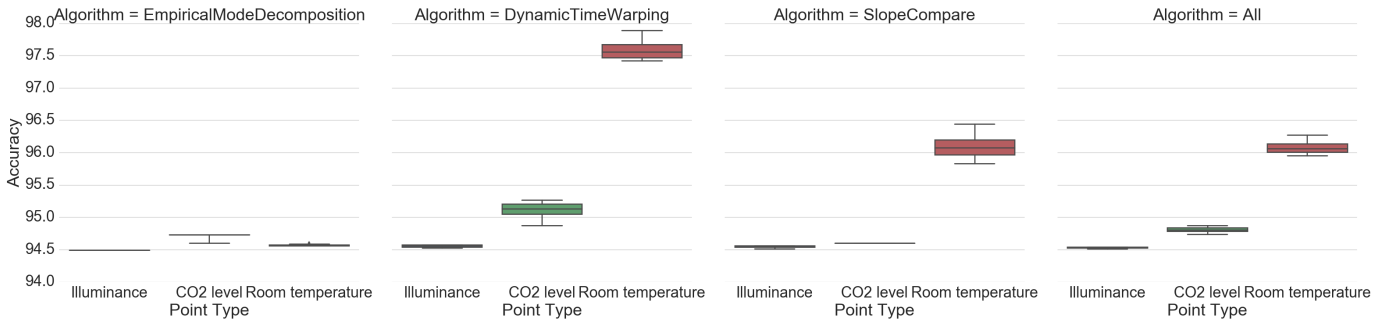

Fig. 5: Results with a combination of 7 (21 validated points) for the three algorithms and All.

0.20. For Temperature the minimum accuracy was 94.56% and a maximum accuracy of 97.89% with a SD of 1.07.

V. Discussion

The results in Fig. 2-5 show that the process of selecting validated points does not effect the results as much as the algorithm and point type in combination. It was expected that the results in Fig. 2 could have a low or random accuracy, due to the fact that the results was generated with only one validated point.

Having more validated points decreased the standard deviation e.g. for Slope Compare from 1.22 to 1.07 using the combination of 7 validated points instead of 1. It is the overall tendency, that the standard deviation has decreased when using more validated points.

Dynamic Time Warping (DTW) has a high score of accuracy regarding points with the point type of room temperature. For DTW and Slope Compare the results for points with point type of room temperature was higher than the two other point types. Only for EMD was the accuracy almost equal for all three point types. This is surprising, due to the fact that room temperature changes slowly, and thereby should fit for EMD. One reason for this, can be found in the raw temperature data, where the temperature is almost steady within a interval of $\pm 2°$C. With an almost steady data stream, the algorithm will only find the slow frequency band of changing between day and night set point.

The three algorithms had success in determining when a point was not similar to another point, but had some difficulties at detecting when two points were similar. This result is not clear from Fig. 2-5, but can be seen in the ratio of true positives compared to the false negatives.

For future work, Metafier should be made full automated, such points will be annotated with metadata when the similarity confidence has reached a certain level. And then we want to have a test, where we use the results from this paper, to extract metadata from another building, located in a different region of Denmark. For algorithms in Metafier, we would like to implement algorithms which can extract information about where the point is located. This could be based on causal relationships or event correlation between points. We would also like to make the algorithms more robust for points where the data stream is almost steady e.g. set points. The current version of Metafier have some difficulties finding set points, due to the implemented algorithms which are using the changes in data to extract patterns. A way for Metafier to know whether a point is a set point, could be based on textual mining of the sparse metadata for a point.

VI. Related Work

Calbimonte et al. [4] have used data mining to annotate and correct metadata regarding a large sensor network. The problem for this large sensor network was that the end users were using different names for the same property, e.g. "temperature", "temp", "t" etc. Calbimonte et al. [4] have focused on a subpart of the metadata problem of automatically generates metadata about the sensor type, whether data comes from a temperature sensor or a humidity sensor. They provide an approach where the raw data stream was split into segments, and those segments was then compared to other sensors, to group sensors of the same type. The type of data streams came from weather stations primarily in the Swiss Alps. This approach has been the inspiration of the approach used in Metafier. For Metafier we have not been able to use segmentation, as Calbimonte et al. [4] used to distinguish between temperature and humidity, due to a larger variety in point types.

Mining data to provide more knowledge of building instrumentation, has been in focus the last couple of years. Software Defined Buildings require a common way of interacting with sensor infrastructures, which can be a cumbersome task. Bhattacharya et al. [3] has shown a way to minimize this time consuming task of annotating sensor infrastructures. In the work of Bhattacharya et al. [3] they have combined active learning and clustering techniques. The data which have been used, are "tags" describing a point in Building Management Systems (BMS). The tags have in first place been created by a human, when the building was build. Those tags have been presented for a domain expert, which has the knowledge to split the tags into meaningful metadata describing the point. Based on a small set of tags, Bhattacharya et al. [3] was able to annotate up to 70% of a buildings BMS points. One problem for this approach is existing errors in tags, that will lead to errors in the learnt model.

Balaji et al. [1] have created the framework Zodiac, which successfully classify sensors with an average accuracy of 98%. Balaji et al. [1] have used BMS tags to extract metadata. They identify the problem of having mistakes in tags. Furthermore they identify the problem of having multiple tags meaning the same or slightly the same. They have used hierarchical clustering in combination with an approach similar to Bhattacharya et al. [3] to extract metadata from the tags. For solving the problem of having mistakes in the tags. They have used clustering of time series data based on four groups of features (Shape, Pattern, Scale, and Texture). To obtain an average accuracy of 98%, they have combined the two approaches, tags, and data streams. The results of the two approaches individual was 94% and 63% for tags and data streams respectively.

We have used an approach like Calbimonte et al. [4], but for building instrumentation. We have shown better accuracy than Balaji et al. [1] for algorithms taking data streams into consideration. Our system is more robust than the approach used by Bhattacharya et al. [3], due to the fact that tags can contain mistakes. Metafier have only been evaluated by three point types, but can easily be extended.

VII. Conclusion

Annotation of metadata from building instrumentation is a time consuming task. We have with Metafier shown algorithms for mining building metadata by data stream comparison. By using several algorithms the system is robust enough to handle

data streams with only slightly similar patterns. We have evaluated Metafier with points and data from one building located in Denmark. We have evaluated Metafier with 903 points, and the overall accuracy with only 3 known examples was 94.71%. Furthermore we found that using DTW for mining points with the point type of room temperature achieved an accuracy as high as 98.13%. Mining building metadata are extremely useful for Software Defined Buildings and for creating the infrastructure for portable building applications. Metafier have only been evaluated by three point types (room temperature, CO_2 level, and illuminance), but can easily be extended.

ACKNOWLEDGMENT

This work is supported by the Innovation Fund Denmark for the project COORDICY (4106-00003B).

REFERENCES

[1] Bharathan Balaji et al. "Zodiac: Organizing Large Deployment of Sensors to Create Reusable Applications for Buildings". In: *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys 2015)*. ACM. 2015, pp. 13–22.

[2] Arka Bhattacharya, Joern Ploennigs, and David Culler. "Short Paper: Analyzing Metadata Schemas for Buildings: The Good, the Bad, and the Ugly". In: *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys 2015)*. ACM. 2015, pp. 33–34.

[3] Arka A Bhattacharya et al. "Automated metadata construction to support portable building applications". In: *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys 2015)*. ACM. 2015, pp. 3–12.

[4] Jean Paul Calbimonte et al. "Deriving semantic sensor metadata from raw measurements". In: *CEUR Workshop Proceedings* 904 (2012), pp. 33–48. ISSN: 16130073.

[5] Stephen Dawson-Haggerty et al. "sMAP: a simple measurement and actuation profile for physical information". In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems* (2010), pp. 197–210.

[6] Romain Fontugne et al. "Strip, Bind, and Search: A Method for Identifying Abnormal Energy Consumption in Buildings". In: *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*. IPSN '13. Philadelphia, Pennsylvania, USA: ACM, 2013, pp. 129–140. ISBN: 978-1-4503-1959-1. DOI: 10.1145/2461381.2461399.

[7] Jayavardhana Gubbi et al. "Internet of Things (IoT): A vision, architectural elements, and future directions". In: *Future Generation Computer Systems* 29.7 (2013), pp. 1645–1660.

[8] Jonathan Holdowsky et al. *Inside the Internet of Things (IoT)*. 2015. URL: https://www2.deloitte.com/content/dam / Deloitte / pe / Documents / technology / Inside % 20The%20Internet%20Of%20Things.pdf.

[9] Emil Holmegaard, Aslak Johansen, and Mikkel Kjaergaard. "Towards a Metadata Discovery, Maintenance and Validation Process to Support Applications That Improve the Energy Performance of Buildings". In: *Proceedings of the 2nd IEEE Workshop on Pervasive Energy Services (PerEnergy 2016)*. Sydney, Australia: IEEE, Mar. 2016, pp. 452–457.

[10] Bo Nørregaard Jørgensen et al. "Challenge: Advancing Energy Informatics to Enable Assessable Improvements of Energy Performance in Buildings". In: *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*. e-Energy '15. Bangalore, India: ACM, 2015, pp. 77–82. ISBN: 978-1-4503-3609-3. DOI: 10.1145/2768510.2770935.